Bern University
of Applied Sciences

generated with DALL·E 3

Engineering Health: The Power of Prompts

Andrew Ellis, Virtual Academy, BFH

Slides are online:

Contents

Understanding Large Language Models

What are LLMs?

- Neural network trained on vast amounts of text data.
- Output a distribution over all possible tokens in a sequence conditional on input.
- LLMs undergo three key stages of training:

1

Pre-training

Trained on large text corpora
Task: next token prediction

→

2

Instruction fine-tuning

Trained on in-/output pairs
Task: next token prediction

→

3

Reinforcement learning
From human feedback

Trained on ratings
Aligns model outputs with human preferences

How do LLMs generate text?

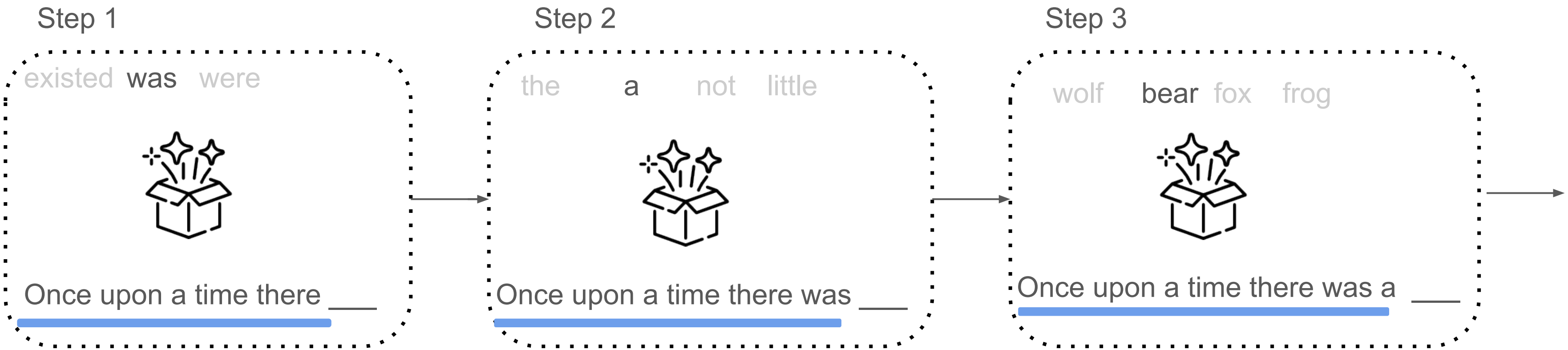
$$P(\underbrace{x_{i+1}}_{}|\underbrace{Context}_{}, \underbrace{Model}_{})$$

- Next token
- The input sequence $x_1, \ldots, x_i$
- Learned model

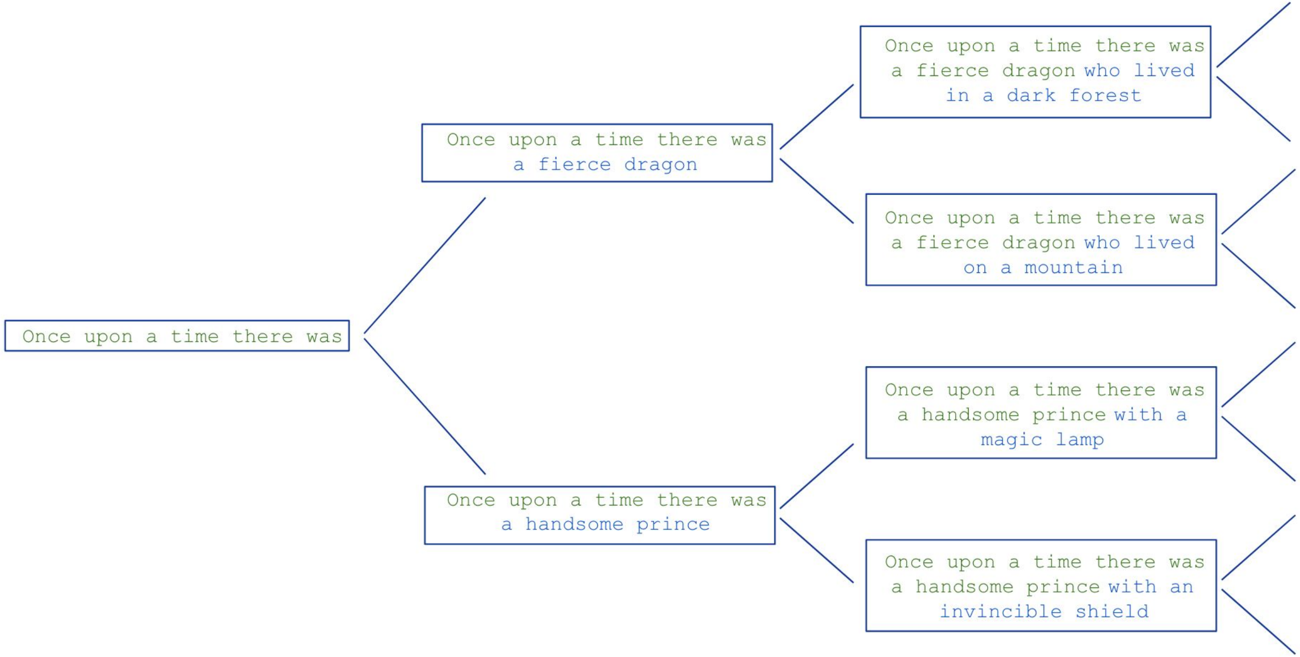
How do LLMs generate text?

$$\underline{P(x_{i+1}}|\underline{Context},\ \underline{Model})$$

Write a story about an animal:



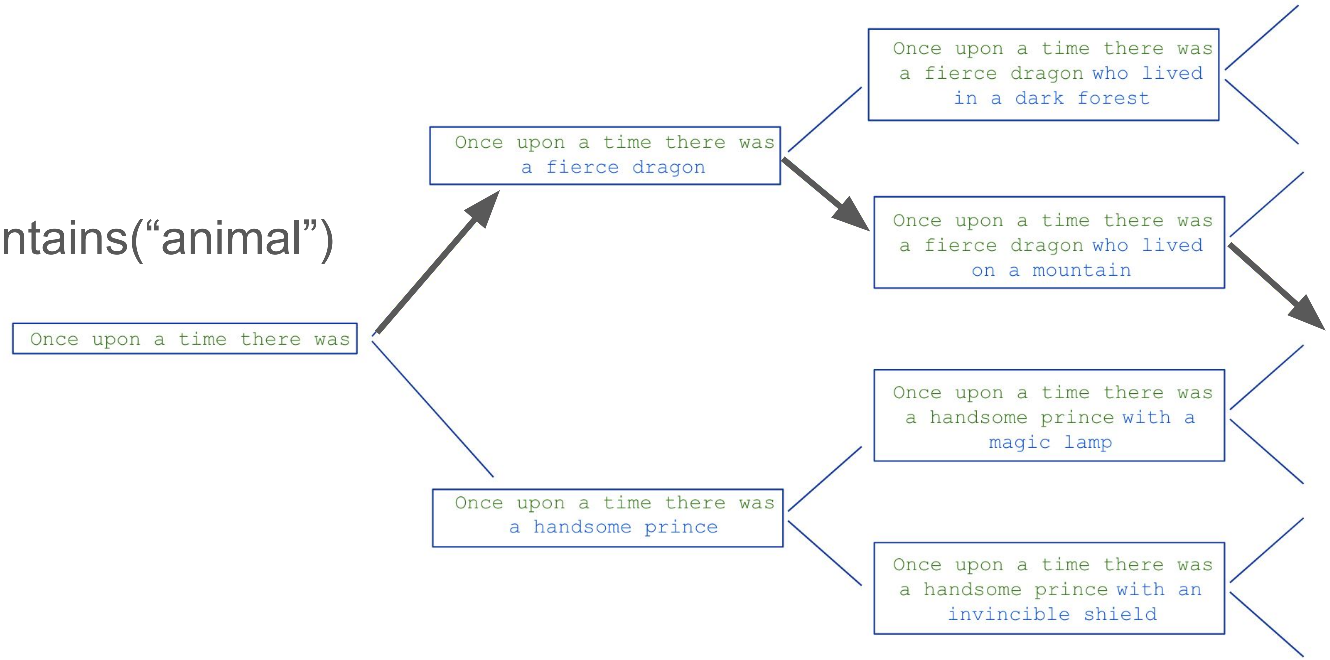
The Space of All Possible Outputs



Shanahan, M., McDonell, K., & Reynolds, L. (2023). Role play with large language models. Nature, 1–6. https://doi.org/10.1038/s41586-023-06647-8

The Space of All Possible Outputs

- contains("animal")



Shanahan, M., McDonell, K., & Reynolds, L. (2023). Role play with large language models. Nature, 1–6. https://doi.org/10.1038/s41586-023-06647-8

What do LLMs learn?

$$P(\underbrace{x_{i+1}}|\underbrace{Context}, \underbrace{Model})$$

- Knowledge is statistical and not grounded in direct experience or sensory input
    - Syntax and grammar
    - Semantics, pragmatics
    - Associations between concepts, facts, and common knowledge

- They can simulate reasoning and planning, but can they actually reason and plan?

Kambhampati, S. (2024). Can Large Language Models Reason and Plan? Annals of the New York Academy of Sciences, 1534(1), 15–18. https://doi.org/10.1111/nyas.15125

AE A farmer needs to transport a wolf, a goat, and a cabbage across a river using a small boat. The boat can carry only the farmer and one other item (either the wolf, the goat, or the cabbage) at a time. However, if left together without the farmer's supervision: The wolf will eat the cabbage. The goat will eat the wolf. The challenge is to figure out how the farmer can get all three items across the river safely. What is the sequence of moves that the farmer needs to make?

Let's think through this step-by-step:

1. First, let's clarify the constraints:
   - The boat can only carry the farmer and one item at a time.
   - The wolf will eat the goat if left alone together.
   - The goat will eat the cabbage if left alone together.
   - The farmer needs to transport all three across safely.
2. Let's solve this puzzle: Step 1: The farmer takes the goat across first. (Wolf and cabbage are safe together on the original side) Step 2: The farmer returns alone. Step 3: The farmer takes the wolf across. (The goat is now alone on the far side) Step 4: The farmer brings the goat back. (Wolf is alone on the far side) Step 5: The farmer takes the cabbage across. (Goat is alone on the original side, wolf and cabbage are on the far side) Step 6: The farmer returns alone. Step 7: Finally, the farmer takes the goat across.
3. The sequence is complete. All items are now on the far side of the river.

🔗 Wikipedia

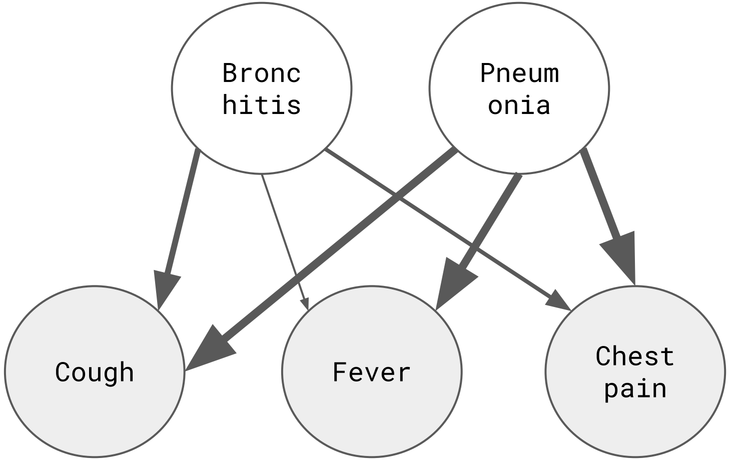
Manhole covers – why are they round?

1. The human body that had to fit through the hole is round.
2. A circle gives a maximum opening for a minimum surface area.
3. Heavy round covers can be rolled rather than lifted into position.
4. They were easier and cheaper to manufacture than square pit covers of the same materials.
5. A circle has a constant diameter and won’t fall into the pit if rotated the wrong way – a tricky problem to remedy if the access cover weighs well over 100kg as concrete and iron covers did.

why-are-manhole-covers-round
See also: Do you think that ChatGPT can reason? [Youtube]

Case Study: Medical Triage Assistant

- A tool capable of reasoning about possible causes of observable symptoms must perform probabilistic reasoning.



IMPORTANT

An LLM cannot do probabilistic reasoning

Prompting: Guiding the Model through its space

$$P(\underline{x_{i+1}}|\underline{Context}, \underline{Model})$$

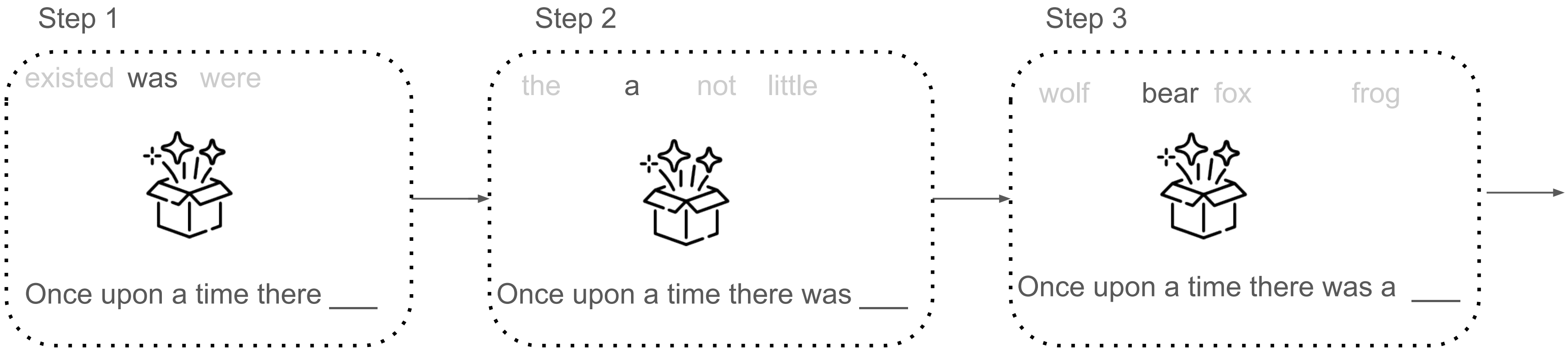
- Prompting guides the LLM along specific paths in its space of possible documents.
- Every token in a prompt reduces the number of potential outcomes, helping the model generate relevant responses.
  - Without a prompt, all outputs are possible.
  - As tokens are added, the range of possible outputs shrinks, making the model's behavior more predictable.

IMPORTANT

Prompting is subtractive

How Prompting Reduces Uncertainty

- Each token conditions the model's next prediction
- With more context, the uncertainty (entropy) decreases, guiding the model towards a more specific output.



The Power of Prompting: Why It Matters

- Controls the behaviour of LLMs, steering them toward relevant outputs.
- Without effective prompting, the full potential of an LLM remains untapped, as it may generate irrelevant or misleading outputs.

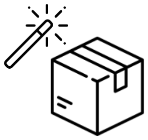
Prompting allows us to:

- Navigate the vast space of possible outputs.
- Achieve more controlled and useful results.

Contexts are combinatorial: we do not know how a model will behave conditioned on all possible contexts.

IMPORTANT

The output is highly contingent on the prompt.



We treat LLMs as black boxes and use engineering approaches to guide their behaviour.

Prompt Engineering:
Level 1

A useful metaphor:

Imagine you are giving instructions to an intern, or a junior assistant.

Role Assignment

Technique: Define a specific role for the AI to adopt.

```
You are an experienced emergency room nurse with over 15
years of experience in patient triage. Your role is to
perform initial assessments of patients based on their
reported symptoms and medical history. You have a calm
demeanor and the ability to quickly prioritize cases based
on severity. In this role, you will categorize patients'
conditions and recommend appropriate next steps.
```

Clear Communication

Technique: Use precise language and specific instructions

```
You are an ER nurse with 15+ years of triage experience.
Your tasks:

Assess patients quickly based on symptoms and medical
history
Categorize conditions by severity
Recommend next steps for treatment

You are calm under pressure and efficient in prioritizing
cases.
```

Provide Context

Technique: Give relevant background information

```
Context: You are working in a busy urban hospital emergency room during
flu season. It's currently 2 AM on a Saturday, and the waiting room is
full. The hospital has been dealing with a recent outbreak of a new
strain of influenza in the community.

Patient Information:
- 45-year-old male
- No known pre-existing conditions
- Not on any regular medications
- Last flu shot was 2 years ago

Given this context and patient information, assess the following
reported symptoms: [Insert symptoms here]
```

Use Examples

Technique: Illustrate desired output with examples

```
Perform a triage assessment based on the patient's symptoms. Format your response
similar to the following examples:

Example 1:
Symptoms: Chest pain, shortness of breath, left arm numbness
Assessment: Emergency
Reason: Symptoms strongly indicate a possible heart attack
Action: Immediate medical attention required. Call for a cardiac team.

Example 2:
Symptoms: Mild fever, sore throat, fatigue
Assessment: Non-urgent
Reason: Symptoms suggest a common cold or mild flu
Action: Rest at home, monitor symptoms, seek medical attention if condition worsens

Now, assess the following patient:
Symptoms: [Insert patient's symptoms here]
```

Specify Output Format: Markdown Table

Technique: Clearly define how you want the information presented

```
Based on the patient's symptoms, create a triage
assessment. Present your findings in a Markdown table with
the following columns: Severity, Primary Concern,
Symptoms, and Recommended Action. Use one of three
severity levels: Emergency, Urgent, or Non-urgent.

The patient has the following symptoms: [Insert symptoms
here]
```

Severity	Primary Concern	Symptoms	Recommended Action
Emergency	Possible Heart Condition or Pneumonia	Chest pain, sweating	Immediate emergency care; call 911 or visit ER
Urgent	Respiratory Infection	Cough, fever	Visit healthcare provider within 24 hours
Non-urgent	General Illness	Cough, mild fever, sweating	Monitor symptoms; rest, fluids, OTC medications

Specify Output Format: Structured List

```
Perform a triage assessment based on the patient's reported symptoms.
Structure your response as follows:
1. Triage Category:
   - [Emergency/Urgent/Non-urgent]
2. Symptoms (in order of severity):
   - [Symptom 1]
   - [Symptom 2]
   - [Symptom 3]
3. Primary Concern:
   - [Brief description of the main medical concern]
4. Recommended Action:
   - [Specific next steps for the patient]
5. Reasoning:
   - [Brief explanation for the triage decision]
Ensure each section is concise and directly relevant to the patient's
condition.
```

Structure Input Using Markdown or XML

Technique: Organize input information in a structured format

E.g. for GPT-4o (using Markdown)

```
# Patient Triage Information

## Patient Details
- **Name**: John Doe
- **Age**: 45
- **Gender**: Male
- **Medical History**: No known pre-existing conditions

## Current Symptoms
1. Chest pain (severity: 8/10)
2. Shortness of breath
3. Left arm numbness
```

Ask an LLM

```
As a language model, how would you proceed when given the
following prompt:
"""
You are an ER nurse with 15+ years of triage experience.
Your tasks:
1. Assess patients quickly based on symptoms and medical
history
2. Categorize conditions by severity
3. Recommend next steps for treatment
You are calm under pressure and efficient in prioritizing
cases.
"""
```

Generate Python Code

Technique: ask an LLM to generate Python code, or in the case of ChatGPT to “use Python”

```
[Insert query here…]

Use Python.
```

Prompt Engineering:
Level 2

Chain of Thought (CoT) Prompting

Explanation: Encourage the model to show its reasoning process*

```
Think through this step-by-step:
1) List the symptoms, 2) Consider possible causes, 3)
Evaluate urgency, 4) Recommend action.
```

or

```
Think step-by-step.
```

* this behaviour has been trained into recent models.

Why Chain of Thought?

Amount of computation is constant per token.

By forcing the LLM to generate more tokens, it will therefore generate more (useful) context, thus making the ‘correct’ result more probable.

Few-Shot Learning

Technique: Provide multiple examples before asking for a new output

```
Input: Chest pain, shortness of breath
Output: Emergency - Possible cardiac event

Input: Mild headache, fatigue
Output: Non-urgent - Rest and monitor

Now categorize: Severe abdominal pain, vomiting blood
```

Structured Output

Technique: Specify a structure for the model's response

```
Provide your assessment in this format:
Severity: [Emergency/Urgent/Non-urgent]
Potential Causes: [List top 3]
Recommended Action: [Specific next steps]
```

Structured Output (bonus tip)

Force the LLM to output valid JSON using the API

```
from pydantic import BaseModel
from openai import OpenAI

class Step(BaseModel):
    explanation: str
    output: str

class MathResponse(BaseModel):
    steps: list[Step]
    final_answer: str
```

```
client = OpenAI()

completion =
client.beta.chat.completions.parse(
    model="gpt-4o",
    messages=[
        {"role": "system", "content":
"You are a helpful math tutor."},
        {"role": "user", "content":
"solve 8x + 31 = 2"},
    ],
    response_format=MathResponse,
)
```

Structured Output (bonus tip)

```
print(message.parsed.steps)

[Step(explanation='Subtract 31 from both sides of the equation to isolate the
term with x.', output='8x + 31 - 31 = 2 - 31'), Step(explanation='Simplify both
sides.', output='8x = -29'), Step(explanation='Divide both sides by 8 to solve
for x.', output='x = \\frac{-29}{8}')]

print(message.parsed.final_answer)

x = \frac{-29}{8}
```

Self-Consistency

Technique: Generate multiple answers, aggregate the responses and select the most popular.

Do this several times independently:

```
Provide three independent
assessments for these
symptoms.
Think step-by-step.

Symptoms:
[insert symptoms here]
```

Provide the responses in a new session:

```
Analyze whether the following
assessments agree with each
other. Give me your expert
assessment based on the
assessments you received.
```

Prompt Engineering:
Level 3

Tree of Thoughts (ToT) Framework for Problem-Solving

Goal: Solve complex tasks by exploring multiple reasoning paths (thoughts).

Steps in the ToT Framework:

- Thought Decomposition: Break down the problem into ‘thoughts’ for reasoning.
- Thought Generation: Generate candidate thoughts.
- Thought Evaluation: Assess each thought’s quality using e.g., scoring.
- Search Algorithms: Use Breadth-First Search (BFS) to explore all possible paths level-by-level.

Global Planning: Incorporate backtracking, lookahead, and pruning to refine the solution path.

Applying ToT and BFS for a Medical Triage Tool

Steps in BFS for Medical Triage:

- Initial State: Patient reports symptoms (e.g., "fever, headache").
- Thought Generation: Generate possible diagnoses (e.g., "flu," "meningitis") using the LLM.
- Thought Evaluation: Evaluate each diagnosis based on severity (e.g., meningitis = high urgency, flu = low urgency).
- Expand and Explore: Generate next steps (e.g., treatment options or recommendations) for each diagnosis.
- Pruning and Backtracking: Keep only the most critical paths (e.g., focus on high-severity conditions).

Outcome: The model systematically evaluates all possible diagnoses and recommends the appropriate level of care.

Why might ToT work?

- Exploration of Multiple Reasoning Paths
  - Better than left-to-right, token-based reasoning.
  - Allows for better search through the problem space.

- Heuristic Evaluation and Pruning
- Backtracking and Lookahead
  - Reconsider earlier decision

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., & Narasimhan, K. (2023). Tree of Thoughts: Deliberate Problem Solving with Large Language Models (No. arXiv:2305.10601). arXiv. https://doi.org/10.48550/arXiv.2305.10601

For more info on ToT, see [Learn Prompting [ToT]]()

Bonus Tip: Keeping Up with Prompt Engineering Research

Technique: Use LLMs to summarize and apply new research

```
Based on the attached research paper on [prompt
engineering technique], write a prompt that would cause an
LLM to behave according to the techniques described in
this paper. Use [topic] as an example.
```

Beyond Prompt Engineering

Prompt optimization

- "Positive thinking" prompts have inconsistent effects across models.
- Chain of Thought (CoT) prompting generally improves performance.
- No universal "best prompt" - effectiveness varies by model and task.
- Automatically optimized prompts often outperform manually crafted ones.
- Optimized prompts can be surprisingly unconventional or eccentric.

Battle, R., & Gollapudi, T. (2024). The Unreasonable Effectiveness of Eccentric Automatic Prompts (No. arXiv:2402.10949; Version 2). arXiv. https://doi.org/10.48550/arXiv.2402.10949

Prompt optimization

Positive thinking:

```
You are an experienced emergency room nurse. Take a deep breath and carefully
assess the following patient's symptoms.
```

CoT:

```
Think through this patient's case step-by-step: 1) List the symptoms, 2)
Consider possible causes, 3) Evaluate urgency, 4) Recommend action.
```

Optimized prompt:

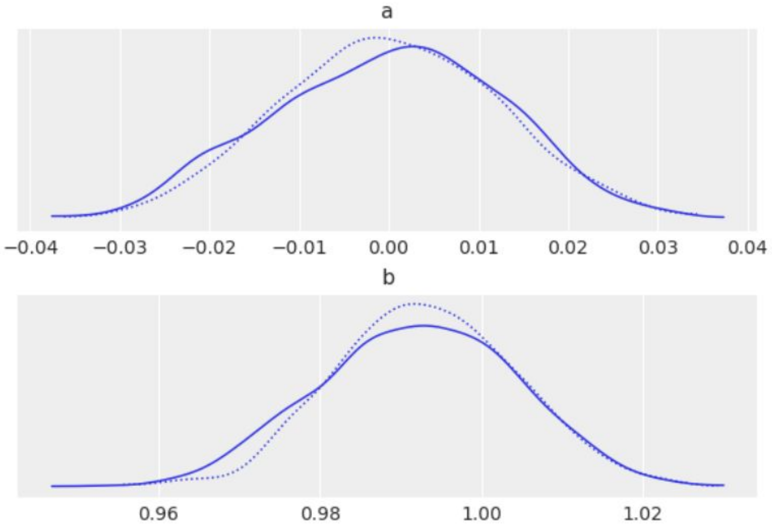
```
The ER is in chaos, Doctor. We need your expertise to navigate this storm of
patients and identify the most critical cases.
```

Battle, R., & Gollapudi, T. (2024). The Unreasonable Effectiveness of Eccentric Automatic Prompts (No. arXiv:2402.10949; Version 2). arXiv. https://doi.org/10.48550/arXiv.2402.10949

Augmenting LLMs

Probabilistic reasoning

- Use LLM to translate natural language utterances to probabilistic programming language*.
- Extract “knowledge” and convert to probability distributions and parameter settings.
- Obtain posterior via MCMC sampling.

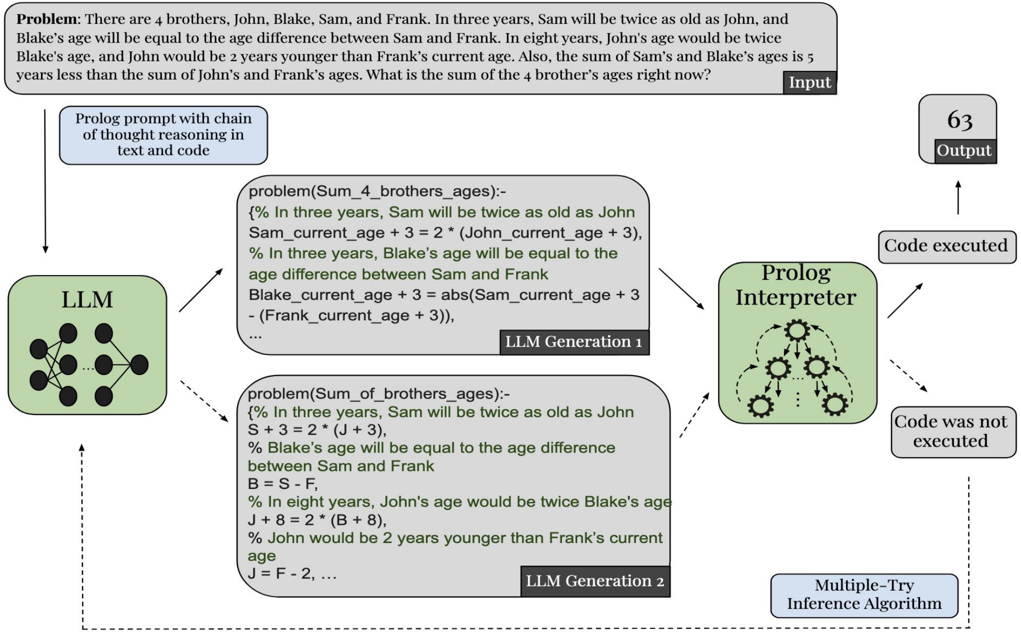


Fremont, D. J., Dreossi, T., Ghosh, S., Yue, X., Sangiovanni-Vincentelli, A. L., & Seshia, S. A. (2019). Scenic: A language for scenario specification and scene generation. Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation, 63–78. https://doi.org/10.1145/3314221.3314633

Li, M. Y., Fox, E. B., & Goodman, N. D. (2024). Automated Statistical Model Discovery with Language Models (No. arXiv:2402.17879). arXiv. https://doi.org/10.48550/arXiv.2402.17879

* PyMC, Stan, Gen, Church, etc.

Deductive reasoning



From: Borazjanizadeh, N., & Piantadosi, S. T. (2024). Reliable Reasoning Beyond Natural Language (No. arXiv:2407.11373). arXiv. https://doi.org/10.48550/arXiv.2407.11373

See also: Poesia, G., Gandhi, K., Zelikman, E., & Goodman, N. D. (2023). Certified Deductive Reasoning with Language Models (No. arXiv:2306.04031). arXiv. https://doi.org/10.48550/arXiv.2306.04031

Take-home message

- Prompting is important because it guides the model through “text space”.
- Prompting range from straightforward to complex.
- Use LLMs to understand prompting.
- Combine manual expertise with automated optimization.
- Use LLMs for things they are good for. Use external tools for everything else.

Bern University
of Applied Sciences

Thank you